Automated Boxwood Topiary Trimming with a Robotic Arm and Integrated Stereo Vision*

Dejan Kaljaca[†,1], Nikolaus Mayer[†,2], Bastiaan Vroegindeweij[1,3], Angelo Mencarelli[4], Eldert van Henten[1], Thomas Brox[2]

***Abstract*— This paper presents an integrated hardware-software solution to perform fully automated robotic bush trimming to user-specified shapes. In contrast to specialized solutions that can trim only bushes of a certain shape, the approach ensures flexibility via a vision-based shape fitting module that allows fitting an arbitrary mesh into a bush at hand. A trimming planning method considers the available degrees of freedom of the robot arm to achieve effective cutting motions. The performance of the mesh fitting module is assessed in multiple experiments involving both artificial and real plants with a variety of shapes. The trimming accuracy of the overall approach is quantitatively evaluated by inspecting the bush pointcloud before and after robotic trimming, and measuring the change in the deviation from the originally computed target mesh.**

I. INTRODUCTION

Garden maintenance is a labor-intensive task which is difficult to automate because of the variation in a natural environment. Some basic gardening tasks have been automated, such as plants watering [1] or small-scale vegetable gardening [2]. Another widespread commercial application of automation in the maintenance of gardens is the robot lawnmower, such as the Bosch Indego robot [3].

However, many gardens include some sort of trimmable plants, like bushes and hedges. These plants require regular trimming to keep their good aesthetic properties. Bush trimming is a demanding task, the automation of which will be of value. For example, elderly or otherwise weakened people are unable to work with the heavy machinery required to trim the plants. Another benefit of automating the trimming task is the reduced labor costs for private and municipal gardens. Finally, such kind of system would be helpful to keep the shape of oddly-trimmed bushes for ornamental purposes, like duck-shaped plants. Pruning such plants without losing the shape requires a high level of precision that is not easy to reach by human gardeners.

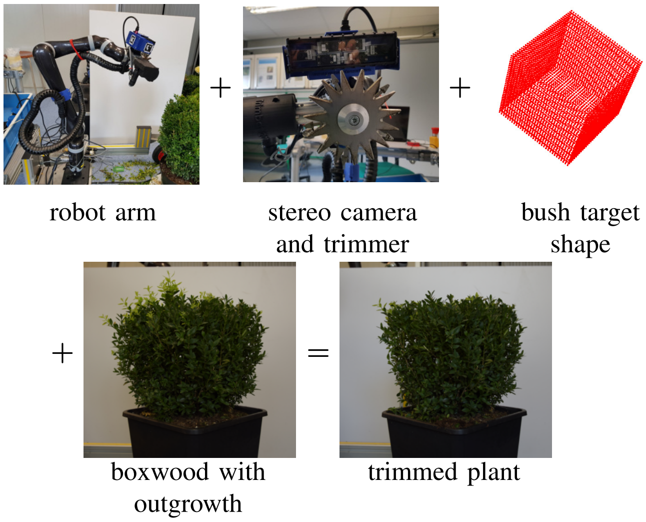


Fig. 1. The presented solution uses arm-mounted vision to scan a target bush. A specified shape is fit into the reconstructed pointcloud. A co-mounted trimming tool cuts the bush using an automatically planned trajectory.

Some machinery already exists for automated bush trimming in tree nurseries. Potatrice Automatica [4] is an example of such a machine. It consists of a cutting frame that can rotate around a bush. The size of the frame has to be manually regulated in a fixed interval, and it can only deal with spherical boxwood bushes.

Bush trimming fits into the research trend of *precision agriculture*. Analogously to other precision agriculture tasks, the automation of bush trimming has to face an unstructured and dynamic environment, which requires advanced perception capabilities. In the last years multiple research works investigated the topics of automated fruit harvesting [5], [6], [7], tree pruning [8], precision orchard farming [9], flower pollination [10], and vineyard maintenance [11].

So far, there is no robot able to trim bushes with variable shapes. To trim differently shaped bushes, the present shape of the bush including outgrown parts must be captured, and a shape fitting algorithm must match the target shape to this reconstruction. Since bush trimming is supposed to work in bright sunlight, 3D sensors based on active light are not suitable, and laser scanners are either limited in their scan coverage, prohibitively expensive, or too bulky to be mounted on the robot arm. RGB-stereo fares well in these aspects. The

*This work was funded by the EU Horizon2020 project ”TrimBot2020”.

[†] These authors contributed equally

[1] Dejan Kaljaca, Bastiaan Vroegindeweij and Eldert van Henten are with the Farm Technology Group of Wageningen University {Dejan.Kaljaca, Eldert.vanHenten}@wur.nl

[2] Nikolaus Mayer and Thomas Brox are with the Department of Computer Science at the University of Freiburg, Germany {mayern,brox}@cs.uni-freiburg.de

[3] Bastiaan Vroegindeweij is also with Livestock Robotics bastiaan@livestockrobotics.nl

[4] Angelo Mencarelli is with Wageningen Plant Research angelo.mencarelli@wur.nl

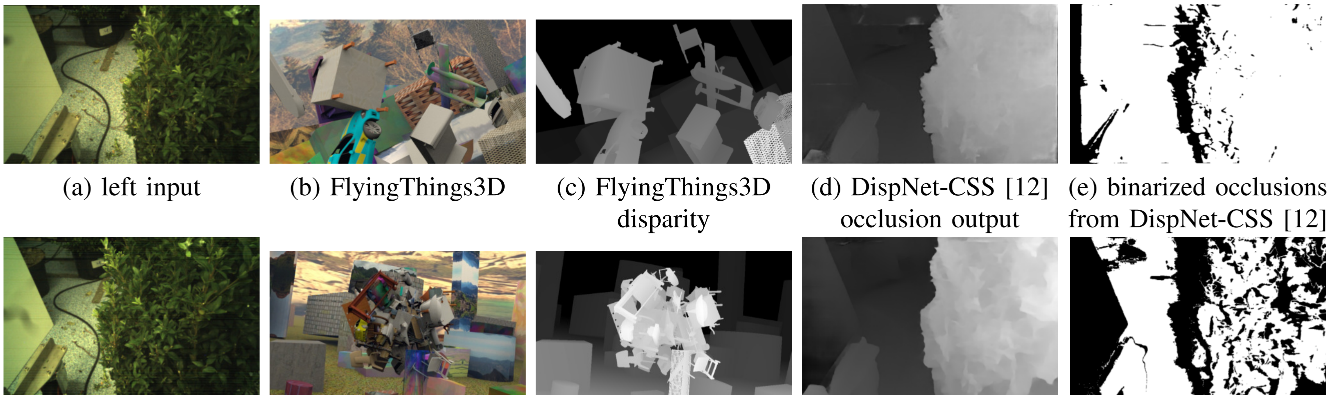

(a) left input
(b) FlyingThings3D
(c) FlyingThings3D disparity
(d) DispNet-CSS [12] occlusion output
(e) binarized occlusions from DispNet-CSS [12]
(f) right input
(g) custom data with tight clustering
(h) custom data disparity
(i) DispNet-CSS with custom data
(j) binarized left-right consistency mask

Fig. 2. DispNet: the original DispNet-CSS (d) [12] was trained on the FlyingThings3D dataset (b,c). To simulate the locally complex plant geometry (a,f), a custom version of the dataset with tightly clustered objects was rendered (g,h). Pretraining on FlyingThings3D and finetuning on the custom dataset yields a network which is better able to resolve fine structures in the bush (i). Instead of the occlusion estimates of the published network (e), left-right consistency is used to remove uncertain and thus probably wrong disparity pixels (j). This check also identifies occlusion areas.

cutting tool must be able to reach multiple locations in a wide workspace and smoothly move along multiple directions in space. Thus, high motion dexterity of a robot arm is needed. The overall approach consists of four processing steps:

1) Scan the target bush using a preplanned robot arm trajectory to acquire an informative bush pointcloud;
2) Fit a user-defined mesh into the reconstruction;
3) Plan a trimming trajectory based on the fitted mesh;
4) Execute trimming with a proprietary trimming tool;

After trimming, the pointcloud of the trimmed plant is compared with the pointcloud before trimming and the original target mesh, to evaluate trimming performance.

II. ROBOT ARM SETUP

A custom test rig was designed and built to test the trimming pipeline. A Kinova Jaco2[13] robot arm was mounted on the test rig. This arm is light-weight, does not need a control box, and consumes little power, which allows mounting it also on a mobile platform (not done for the experiments in this paper). The 6 rotational joints of the arm provide the dexterity needed for scanning and trimming trajectories.

On the last joint of the arm, the trimming tool [14] and a stereo camera[1] were mounted as a rigid unit. Extrinsic hand-eye calibration was done using HALCON [15]; stereo-extrinsic and intrinsic calibration was done using Kalibr [16].

An overview of the hardware setup is shown in Fig. 1. The software for the experiments was based on the Robot Operating System (ROS) [17].

III. SCANNING + POINTCLOUD MERGING

A. *DispNet for stereo vision*

The arm-mounted RGB stereo camera was used to acquire image data and corresponding depth maps. The depth map was computed with DispNet [12], which is a deep network trained on synthetic data from FlyingThings3D [18] based on objects from ShapeNet [19], to yield dense disparity maps at interactive frame rates.

DispNet assumes that the images are rectified, and performs correspondence search along horizontal scanlines. Thus, the images are undistorted and rectified before being fed into the network.

The stacked ”CSS” architecture from Ilg et al. [12] was used. It consists of a stack of three encoder-decoder networks, with the first network including an explicit correlation layer and warping layers between each subnetwork. The network was pretrained on the FlyingThings3D dataset [18] following the original stagewise approach [20], [12]: first the first ”C” stage was trained in isolation; then weights at this stage were frozen and the second ”S” stage was trained to process the outputs of ”C”; finally both were frozen and the third ”S” stage was trained on the outputs of ”CS”. These three training steps used the S_{short} schedule [20].

After such pretraining, this network was finetuned on a custom version of FlyingThings3D in order to match the specific challenges of the task and the used camera. To simulate the challenges of plants with their fine-detailed branch structures and many occlusions, many ShapeNet objects were tightly clustered together. Moreover, the images were rendered with a simulated Bayer color filter to reduce the domain gap between train and test data [21]. Samples from this dataset and from FlyingThings3D are shown in Fig. 2. Network finetuning used the S_{fine} learning rate schedule [20]. Both pretraining and finetuning used data augmentation in color and geometry [21].

While DispNet yields disparity maps with state-of-the-art accuracy, there are still some inaccurate estimates, especially close to depth boundaries. These inaccuracies cause outliers in the pointcloud. Therefore, the disparity maps were cleaned up via a left-right-right-left consistency check, i.e., not only the left-right disparity $d_{\mathbf{L}}$ of a pair $\mathbf{LR}$ is computed, but also the right-left disparity $d_{\mathbf{R}}$: by rotating both input images by

[1] Resolution $= 752\times480$, diagonal FoV $\approx 68^\circ$, baseline $\approx 5\,\text{cm}$

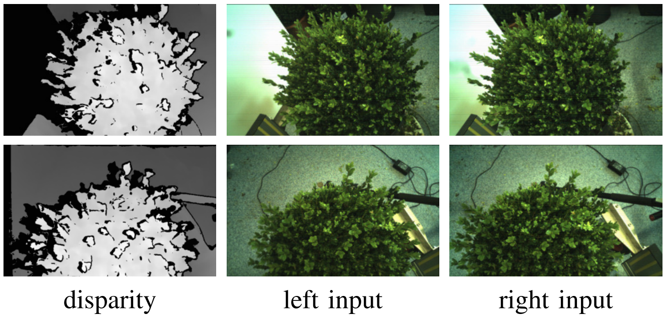

disparity left input right input

Fig. 3. DispNet can resolve fine structures of an outgrown boxwood bush, such as twigs sticking out at the side or towards the camera. Such detailed perception and dense coverage are needed when processing real plants.

180° and swapping them, the network sees $\rotatebox{180}{\textbf{LR}}$ and computes a disparity map $d_{\rotatebox{180}{\textbf{R}}}$ for the "left" view $\rotatebox{180}{\textbf{R}}$. Consequently, $d_{\mathbf{R}}$ is obtained from $d_{\rotatebox{180}{\textbf{R}}}$ by another 180° rotation. If for a pixel (x, y) in $d_{\mathbf{L}}$, the left-right disparity $d_{\mathbf{L}}(x, y)$ does not agree with the corresponding right-left disparity $d_{\mathbf{R}}(x - d_{\mathbf{L}}(x, y), y)$, the point is marked as invalid in the disparity map (shown as black pixels in Fig. 3.

Additionally, a gradient magnitude filter (to remove any remaining "depth curtain" artifacts) and a 5×5 median filter are applied to the disparity map.

The capability of the DispNet-CSS architecture to directly estimate occlusion areas and depth boundaries [12] was not exploited because occlusions are not the only source of inaccuracies in the disparity map. Lens flare, spurious false matches, and oversmoothing due to repetitive or featureless scene structure also contribute to this. Using the left-right consistency check allows handling all of them reliably. Fig. 2 compares the custom-finetuned network and left-right consistency masking against a pretrained DispNet with built-in occlusion estimation [12].

B. Scanning

A single viewpoint can only provide limited information due to complex self-occlusions (see e.g. Fig. 3; a branch sticking out can be perceived, but not the bush surface behind it). As a consequence, fitting target shapes into a pointcloud from a single view proved to be unreliable, especially with significantly outgrown plants (see Fig. 4). Therefore, the pointcloud was assembled from depth maps at multiple viewpoints.

The robotic arm performed a predefined scanning motion, shortly pausing at intermediate poses to avoid motion blur when recording an image pair. For each of these poses a disparity map is computed. Since DispNet runs at 4.5 fps, the length of the pause is determined by the acceleration limits of the arm and not by DispNet.

The disparity maps were converted to pointclouds in a global space, and fused by simply accumulating all points from all views using the poses reported by the arm. To make the system robust to inaccuracies in reported arm poses, experiments with ICP alignment of individual views' pointclouds were conducted, but the arm proved to be sufficiently precise. The procedure was rather limited by the inaccuracy of the disparity maps. To make the shape fitting more performant, the composite pointcloud was spatially subsampled after each new view integration.

IV. Shape Fitting

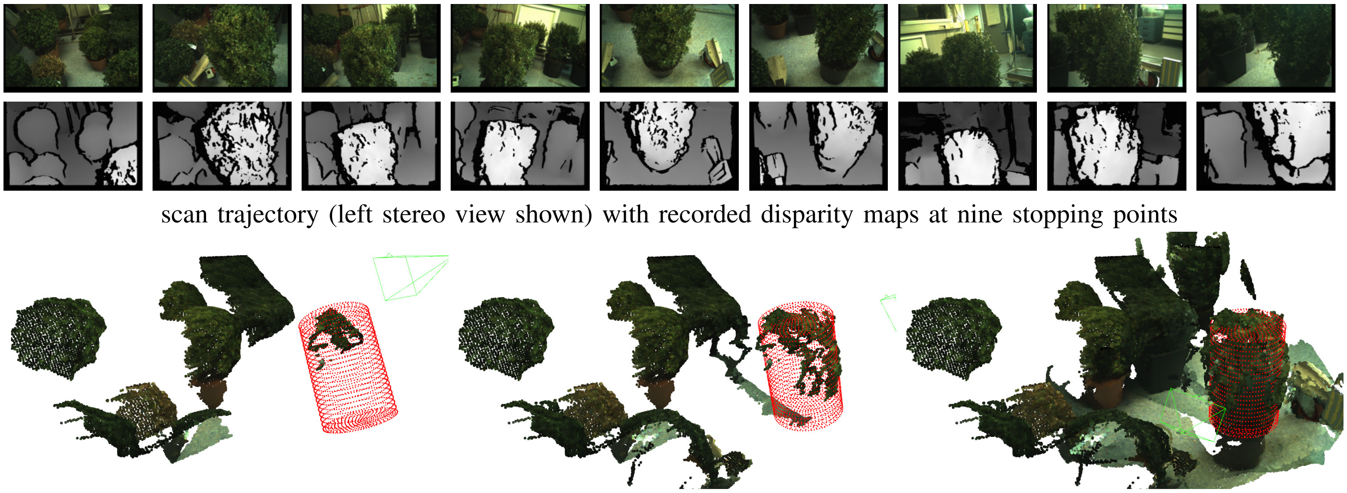
The purpose of trimming is to cut the bush to a predefined target shape. Fitting the target shape into the observed pointcloud was done using a variant of *Trimmed ICP* [22]: instead of ordering the point correspondences by distance and then discarding a fixed ratio, all correspondences with distances that exceed a fixed threshold are ignored. This ensures the coverage of the scene-scanning trajectory to not directly affect the fitting result: a clean spherical bush leads to a good sphere fitting even from a single view with less than 50 % correspondence coverage, whereas the experiments on outgrown bushes required a much larger coverage for a good fit (see Fig. 4 and Fig. 8). The fitting process was run online during bush scanning and is fast enough to not limit the speed of the scanning process. The bush position was initialized using the pixel closest to the camera in the first obtained disparity map.

After fitting, the mesh for the target shape was shrunk by a user-defined amount (e.g. "cut back by 2 cm") and sent to tool-motion planning and execution.

V. Trimming Motion Planning

In an ideal bush trimming motion, the trimming point of the trimming tool should be the one closest to the bush, with other parts of the tool further away. The movement of the trimming tool would then be such that the trimming point is placed most upfront in the direction of movement, thereby grasping and cutting branches before they come in contact with other parts of the trimming tool. As the current trimming tool could be used for omnidirectional trimming, in these first tests such constraint was relaxed and side cutting motions were allowed as well.

Trimming trajectories were planned according to the procedure described in [23]: first, the triangles belonging to the target mesh are clustered according to spatial closeness and surface normal similarity, thus defining a limited amount of tool poses to be traversed for bush trimming. Then, multiple arm configurations matching the desired tool poses are queried to an inverse kinematics solver based on the arm geometric model. Poses that turn out to be unreachable (e.g. too close or too far away, or too close to the ground for safe approach) are removed. Finally, the desired arm configurations schedule was computed by approximately solving a Generalized Traveling Salesman Problem on the graph having the retrieved arm configurations as nodes. The planning cost function was defined to ensure regular arm movements in the cutting phase, thus allowing an accurate and smooth trimming action. For this, several motion optimality measures can be used, such as path length or number of turns [24]. A suitable solution to the Generalized Traveling Salesman Problem was retrieved using the Ant Colony Optimization method [25]. The Edge Distance Function (EDF) between each pair of candidate configurations was defined as such:



scan trajectory (left stereo view shown) with recorded disparity maps at nine stopping points

one viewpoint

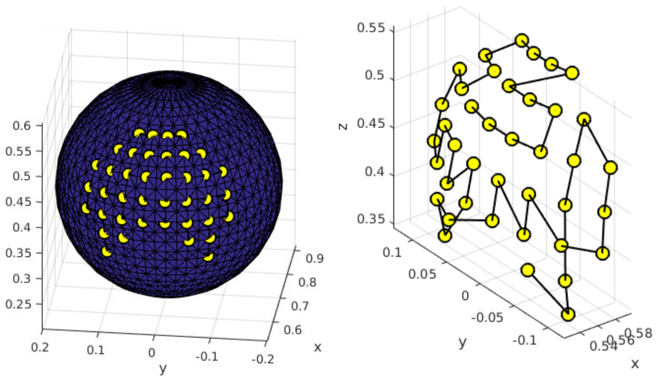
two viewpoints

nine viewpoints

Fig. 4. Multiview scanning and fitting: with only a few viewpoints that do not sufficiently cover an unruly target object, shape fitting is unstable. More views' information progressively improves the result; in this case, nine views were needed for a stable fit. These nine views are shown in the **top row**. Note that insufficient visibility of the plant is not the reason why early fitting fails; e.g. the second and forth view include almost the entire bush.

1) If the distance between the target tool poses is lower than a threshold Δ_{max}, and the corresponding tool movement direction is not downward, the EDF is defined as $EDF(q_i, q_j) = \Sigma_{k=1}^{N_J} |q_{i,k} - q_{j,k}|$, where N_J is the number of joints and $q_{i,k}$ is the k-th joint position of the arm configuration q_i.
2) Elsewhere, the EDF is assigned a threshold value. Such a value indicates the need of a non-cutting arm transition between the given target tool poses (i.e. a retraction and a re-approaching motion). As the amount of non-cutting transitions should be kept small for the sake of efficiency in task time and energy consumption, the threshold value is heuristically set as 100, which is significantly higher than the average arm configuration distance.

The goal is to search for a path $q_1, ...q_{N_P}$ minimizing the cost function $\Sigma_{i=1}^{N_P-1} EDF(q_i, q_{i+1})$, where N_P is the number of tool poses.



reachable surface area

computed tool path

Fig. 5. Example of planned trimming tool path on a sphere mesh.

VI. EXPERIMENTAL APPROACH

A. Data collection

To test the trimming pipeline, several bush shapes were used. This included artificial, real pretrimmed bushes, and outgrown bushes, with spherical, cubical, and cylindrical shape. Furthermore, some composite shapes were added to test fitting performance on non-symmetrical objects. A sample of the bushes used in the evaluation is shown in Fig. 6.

The following protocol was applied for each bush:

1) Place bush in the test rig;
2) Execute scanning trajectory and store acquired data;
3) Run shape fitting on stored data.

A sphere-shaped and a cube-shaped bush were selected for trimming. For those bushes, the following trimming steps were executed in addition:

4) Plan trimming path;
5) Execute trimming path;
6) Repeat scanning step 2 above;
7) Compare observations before and after trimming.

B. Evaluation of shape fitting

To evaluate shape fitting, the distances of 3D points in the scene reconstruction to the fitted ideal shape were computed and sorted into a histogram. To account for the variation in 3D reconstruction density due to incidental factors such as lighting, data was normalized with respect to the number of points in the point cloud. The distribution of points in the histogram then indicates the quality of fitting: if the majority of points is close to the fitted shape, then the shape is both the correct geometry and size, and also well fitted into the observation. If the bush is already well-shaped, all distances should be close to zero. Branches of an outgrown bush will lead to outliers in the histogram, but the majority of distances should be small for a well-fitted shape. As shown in Fig. 7,

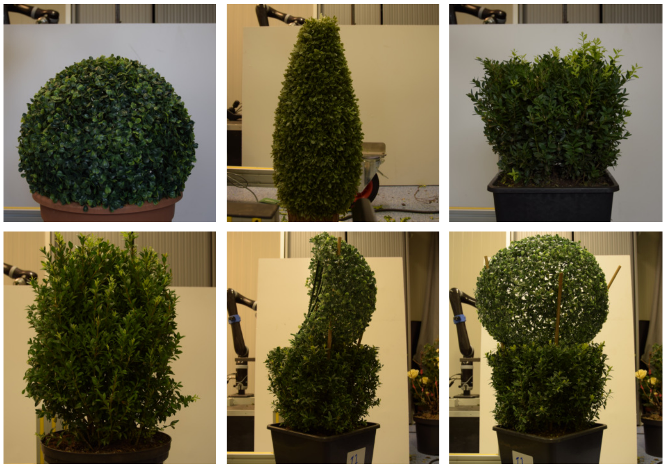


Fig. 6. A sample of the bushes used in evaluation. The sphere (top-left) and pinecone (top-centre) are artificial bushes, the cube (top-right) and cylinder (bottom-left) are real bushes, and the composite shapes (bottom-centre and bottom-right) are a mixture of real and artificial bushes

the points belonging to the bush can be clearly separated from the scene background by simple distance thresholding. Thus, the distance range for evaluation was capped at 15 cm from the fitted shape.

C. Evaluation of trimming

Trimming evaluation used the same distance histogram measure as the evaluation for shape fitting. If on average more points are closer to the target surface after trimming than before, trimming was considered successful. Fig. 1 shows an example of this: the outgrowth of the cube bush is removed.

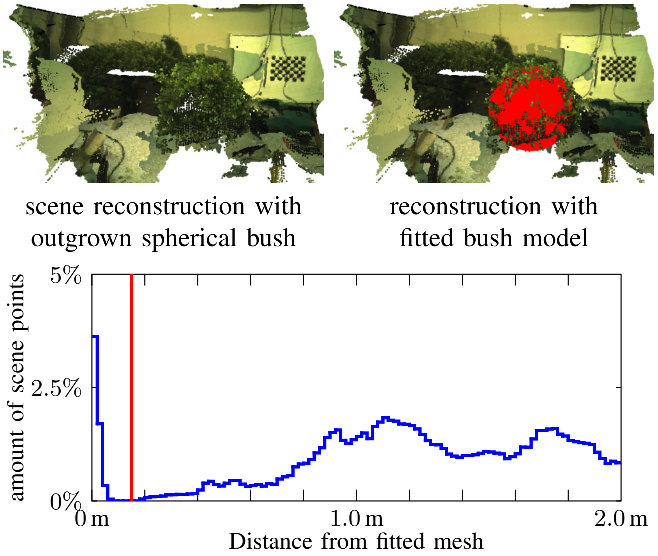


Fig. 7. Histogram plot of scene point distances to a fitted sphere mesh. The reconstruction of the bush itself is clearly separated from the rest of the scene; for evaluation, distances beyond 15 cm (red line) are ignored. Bin width = 2 cm.

VII. Results and Discussion

Results are presented step-by-step through the pipeline, evaluating the quality of shape fitting using the alignment of the scene reconstruction point cloud against the fitted ideal shape of a bush. First, fitting was tested on artificial bushes with well-defined shapes, using only one scanning viewpoint. This serves to show that DispNet produces correct depth information, and that the fitting procedure is not generally limited by low scan coverage. Next, more complicated shapes are fitted. These require scanning from multiple viewpoints, due to self-occlusions or bush size. With robust multiview fitting established, real outgrown boxwood bushes are finally scanned and trimmed.

Fig. 7 shows a complete reconstruction pointcloud and distribution of point distances to a fitted shape. There is a clear separation between points that are part of the bush and points in the background; the rest of this section thus ignores all points more than 15 cm away from the fitted shape.

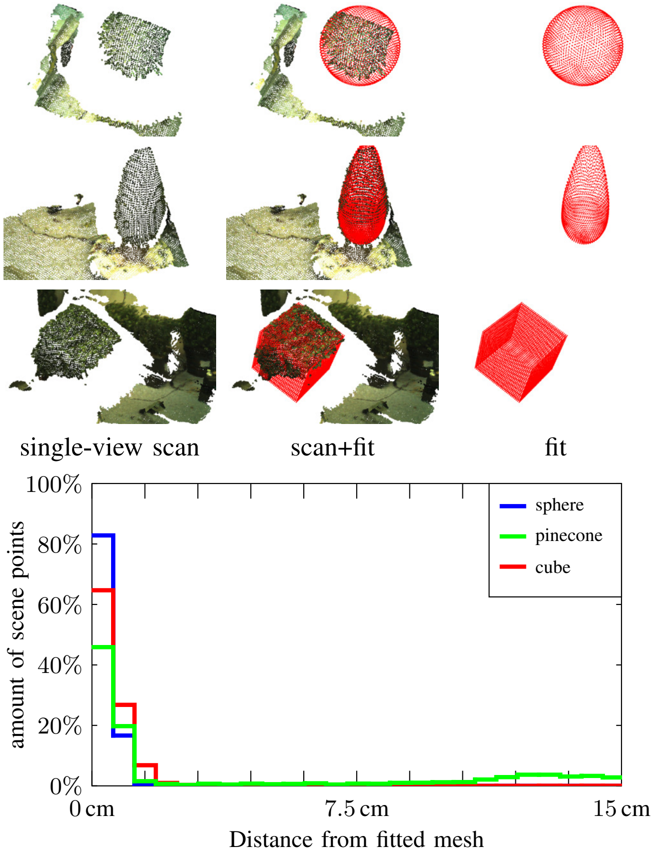


Fig. 8. Fitting evaluation showing the distance of pointcloud points with respect to the resulting mesh of artificial bushes: sphere, pinecone-shape, and cube. A single viewpoint provided sufficient information for a good fit, as seen in the "scan+fit" column where the mesh overlapped the bush pointcloud. Histograms are normalized over all reconstructed scene points placed at most 15 cm from the target surface. Histogram bins are 0.6 cm wide.

A. Scanning and shape fitting

1) Artificial bushes: The ideal case for fitting is a smooth shape that is already close to the target shape. Fig. 8 shows that for artificial bushes with clear shape, a single viewpoint was already sufficient: for an artificial sphere, pinecone, and cube, the majority of reconstructed scene points was either within 0.6 cm from the fitted shape (i.e. in the first histogram bin) or further away than 15 cm (i.e. not a part of the target object). Within these shapes, the pinecone yielded the worst quantitative result: this shape was modelled by hand which makes it hard to get its dimensions exactly right. The large-distance outliers for the pinecone were due to this reconstruction and from including artefacts from the planting pot and the adjacent floor.

2) Real plants and complex shapes: With more complex shapes and real plants (which do not perfectly match the target shape), a single viewpoint was not enough to provide sufficient information for a good shape fit. Fig. 4 shows that for this cylinder shape even two views' worth of data did not lead to a good fit; a stable result was only reached after nine views had been fused. The scanning trajectory producing these views was used for all multiview fitting and trimming experiments. In Fig. 9, a cylinder-shaped boxwood bush and an artificial complex shape (made up of a cube and a hemisphere on top) were evaluated. The final pose of the target shape conformed to the true position and orientation, but noise in the depth data and irregularities in the scanned objects decreased the fitting accuracy. As result, the histogram is less focused compared to the simple artificial shapes above, although most points stay within 5 cm of the fitted mesh.

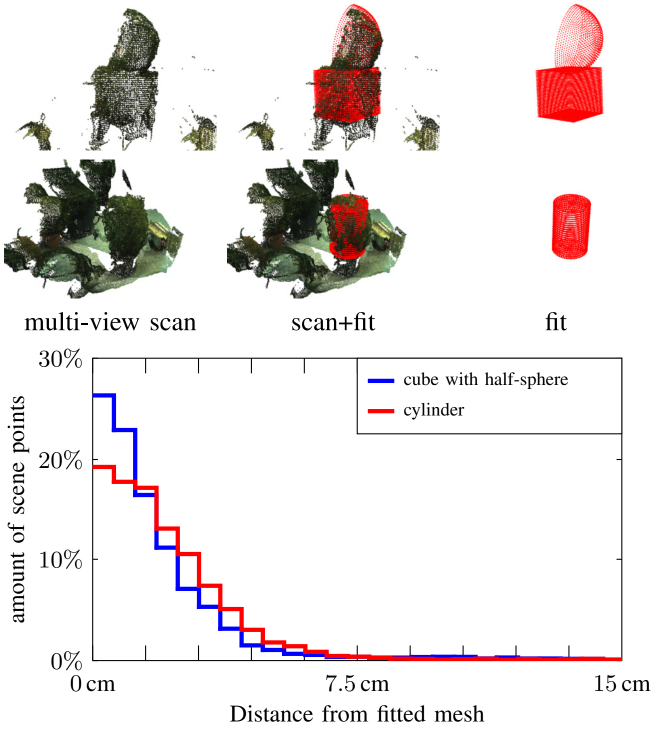


Fig. 9. Fitting evaluation of a real cube bush with an artificial hemisphere fixed on top, and a real cylinder-shaped bush. As highlighted in Fig. 4, multiple viewpoints needed to be fused for these shapes to fit well. The cylinder bush did not conform well to its target shape, as it was slightly tilted. This is visible both in Fig. 4, as well as from the larger spreading of the points in the histogram. Plots as in Fig. 8.

B. Mesh-based bush trimming

An example of a trimming plan based on a target spherical mesh is shown in Fig. 5. During plan execution, the trimming tool was correctly sent to its target poses and proved capable of cutting outgrown branches. To evaluate trimming performance, the histogram of scene point distances from the fitted mesh before trimming is compared to the corresponding histogram after trimming. For the sphere bush, applying the initially fitted mesh on the trimmed pointcloud showed that the plant had slightly changed its position. Thus, fitting a new mesh on the trimmed bush turned out to be necessary in order to extract meaningful results about the trimming effect.

Figures 10 and 11 show the histograms of point distances before and after trimming, respectively, for a cube-shaped bush and a sphere-shaped bush. Both figures show that point distances change after trimming, by increasing the percentage of points having a low distance from the target mesh. At the same time, the percentage of points being more than 3 cm away from the mesh reduced from 30 % to 20 % for the cube and from 35 % to 26 % for the sphere, indicating that these points were most affected by the trimming operation. From visually observing the experiments, it was found that not all motions conformed well to the desirable behaviour as described in section V. Also, it was observed that sometimes branches were pushed aside instead of being trimmed. As result, the trimming performance might have been lower than one would have expected, although it still demonstrates the functionality of the concept.

Improved trimming performance is expected by better matching the planner behaviour to the desired cutting paths and updating the trimming tool design to avoid branches being pushed aside. In both cases, the trimming results should approach the target mesh more closely, such that after trimming the percentage of pointcloud points close to the target mesh will increase with respect to the values observed in these experiments. Another improvement step was already performed when trimming the sphere as shown in Fig. 11. After executing the trimming trajectory once, the bush was rotated by 90° and trimmed again. The comparison between the resulting three histograms shows that the point distribution became closer to the target mesh after each trimming execution. As can be observed from the images, outgrowth was removed on the left and right side of the bush, but some branches remained. Currently, trimming is limited by how much of the bush surface can be reached by the robot arm. Bush rotations were done using a rotating table, but for example the top of the sphere remained unreachable. Placing the arm on a mobile platform increases flexibility in arm placement, thereby allowing to trim from multiple positions. Such an approach would also allow to trim bushes in a garden setting.

VIII. CONCLUSION

A working pipeline for autonomous bush trimming with a robot arm equipped with a vision sensor was presented. The robot arm scanned a scene to produce a composite

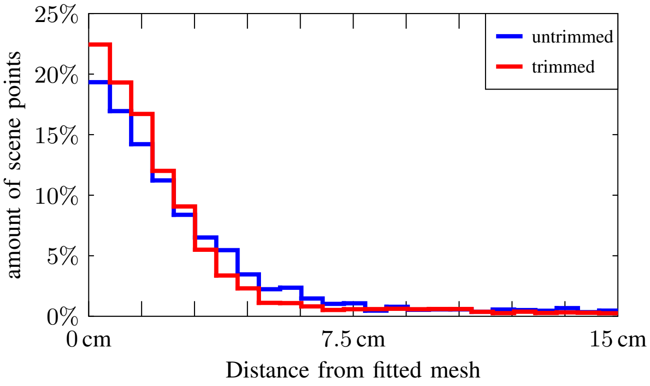


Fig. 10. Trimming evaluation of the cube bush with outgrowth showcased in Fig. 1. After trimming, more points are closer to the target shape than before. Plots as in Fig. 8.

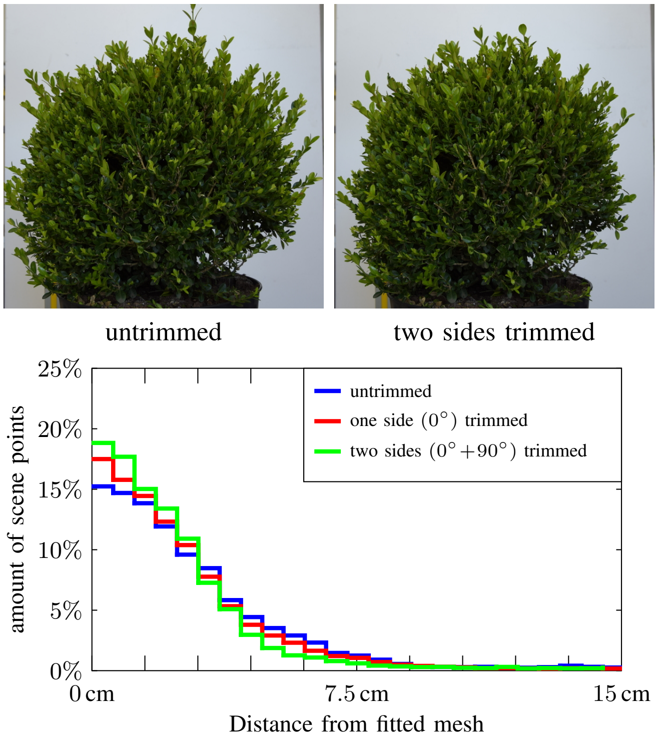


Fig. 11. Trimming evaluation of a spherical bush with significant outgrowth. After one trimming step, the bush was rotated by 90° and re-scanned. A new trimming trajectory was planned and executed, after which the bush was scanned once more. The images on top show that outgrowth was removed on the left and right sides, but not all branches were caught by the trimmer. The histogram shows that each trimming sequence improved the point distribution, bringing more points closer to the desired shape. Plots as in Fig. 8.

scene pointcloud, into which a predefined target mesh was fitted. A coverage planning algorithm then used this mesh to generate a bush trimming trajectory. In the final step, the trimming trajectory was executed and trimming performance was assessed. The presented approach was designed to effectively deal with fitting and trimming of arbitrary plant shapes. The performed experiments show that multiple shapes were effectively handled.

It was shown that for artificial plants the fitting accuracy was good, as over 50 % of the bush points was less than 0.6 cm from the fitted mesh. When fitting real plants, this percentage decreased due to the irregularity of the bush. Nevertheless the majority of the points resulted to be less than 3 cm from the target.

Concerning trimming, evaluation was performed by comparing the distribution of scene point distances from the fitted mesh before and after the execution of the trimming trajectory based on the initially computed mesh. For a sphere-shaped and a cube-shaped bush, the percentage of points further than 3 cm from the target was decreased by $25-33$ % (relative) after trimming. This shows that the robot is able to effectively trim the bush using this approach.

As result, this work is a significant step towards a fully automated, self-evaluating robotic trimming system, which is able to take decisions based on the outcome of its own trimming actions.

ACKNOWLEDGMENT

The authors would like to thank Jochen Hemming of Wageningen Research for his support in executing the hand-eye calibration and setting up the experiments.

REFERENCES

[1] Automatic irrigation. Accessed on 01/03/2019. [Online]. Available: https://www.gardena.com/nl/producten/geleiding/automatic-irrigation/

[2] Open-source cnc farming. Accessed on 01/03/2019. [Online]. Available: https://farm.bot/

[3] Bosch Indego lawnmower. Accessed on 01/03/2019. [Online]. Available: https://www.bosch-garden.com/gb/en/garden-tools/homepage/

[4] Potatrice Automatica. Accessed on 01/03/2019. [Online]. Available: https://www.youtube.com/watch?v=P5X-N2b0H9c

[5] E. van Henten, J. Hemming, B. van Tuijl, J. Kornet, and Bontsema, "An Autonomous Robot for Harvesting Cucumbers in Greenhouses," *Autonomous Robots*, vol. 13, no. 3, pp. 251–258, 2002.

[6] N. Correll, N. Arechiga, A. Bolger, M. Bollini, B. Charrow, A. Clayton, F. Dominguez, K. Donahue, S. Dyar, L. Johnson, H. Liu, A. Patrikalakis, T. Robertson, J. Smith, D. Soltero, M. Tanner, L. White, and D. Rus, "Building a distributed robot garden," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 1509–1516, 2009.

[7] C. W. Bac, E. J. van Henten, J. Hemming, and Y. Edan, "Harvesting robots for high-value crops: State-of-the-art review and challenges ahead," *Journal of Field Robotics*, vol. 31, no. 6, pp. 888–911, 2014. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21525

[8] S. Paulin, T. Botterill, J. Lin, X. Chen, and R. Green, "A comparison of sampling-based path planners for a grape vine pruning robot arm," *International Conference on Automation, Robotics and Applications*, pp. 98–103, 2015.

[9] A. Gasparri, G. Ulivi, N. Bono Rossello, and E. Garone, "The H2020 project Pantheon: precision farming of hazelnut orchards (extended abstract)," in *Convegno Automatica*, Florence, Italy, Sep 2018. [Online]. Available: http://pantheon.inf.uniroma3.it/images/Publications/7.pdf

[10] N. Ohi, K. Lassak, R. Watson, J. Strader, Y. Du, C. Yang, G. Hedrick, J. Nguyen, S. Harper, D. Reynolds, C. Kilic, J. Hikes, S. Mills, C. Castle, B. Buzzo, N. Waterland, J. Gross, Y. Park, X. Li, and Y. Gu, "Design of an autonomous precision pollination robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 7711–7718.

[11] GRAPE Project. Accessed on 01/03/2019. [Online]. Available: http://www.grape-project.eu

[12] E. Ilg, T. Saikia, M. Keuper, and T. Brox, "Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or

scene flow estimation," in *European Conference on Computer Vision (ECCV)*, 2018.

[13] Kinova Robotics. Accessed on 01/03/2019. [Online]. Available: http://www.kinovarobotics.com

[14] B. van Tuijl, A. Tielen, A. Mencarelli, and J. Hemming, "Structured design of a novel end-effector for a bush trimming robot," in *European Society of Agricultural Engineers (EurAgEng) Conference, Ageng 2018*, 2018.

[15] HALCON. Accessed on 01/03/2019. [Online]. Available: https://www.mvtec.com/products/halcon

[16] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 473–480.

[17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[18] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, arXiv:1512.02134.

[19] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository, Tech. Rep. ArXiv preprint arXiv:1512.03012, 2015.

[20] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox, "What makes good synthetic training data for learning disparity and optical flow estimation?" *International Journal of Computer Vision*, vol. 126, no. 9, pp. 942–960, Sep 2018, https://arxiv.org/abs/1801.06397.

[22] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Object recognition supported by user interaction for service robots*, vol. 3. IEEE, 2002, pp. 545–548.

[23] D. Kaljaca, B. Vroegindeweij, and E. van Henten, "Coverage Trajectory Planning for a Bush Trimming Robot Arm," *submitted to Journal of Field Robotics*.

[24] J. Hess, G. D. Tipaldi, and W. Burgard, "Null space optimization for effective coverage of 3D surfaces using redundant manipulators," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1923–1928, 2012.

[25] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem." *Bio Systems*, vol. 43, no. 2, pp. 73–81, 1997.